UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/807,833	03/24/2004	Theodore C. Goldstein	2095.001000/P3125US1	4323

23720 7590 07/19/2007
WILLIAMS, MORGAN & AMERSON
10333 RICHMOND, SUITE 1100
HOUSTON, TX 77042

EXAMINER	
WU, JUNCHUN	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
07/19/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/807,833	GOLDSTEIN ET AL.	
	Examiner	Art Unit	
	Junchun Wu	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 March 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-53* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-53* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-53 are pending in this application.

Claim Objections

2. Claim 12 is objected to because of the following informalities:

- Claim 12 contains a surplus word, in line 4, "by a waypoint from a standard input open", the "open" should be deleted.

Appropriate correction is required

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims are 1-12, 15-24, 27-33, and 38-53 are rejected under 35 U.S.C. 102(b) as being anticipated by DaSilva et al. (US Pat. No. 6, 493,868 B1, hereinafter "DaSilva").

5. Per claim 1

DaSilva discloses

- A method for use in developing a program, comprising compiling at least a portion of a source code program defined by a waypoint during the editing of the source code program (col.2 lines 50-55; setting a breakpoint to stop the execution of a program).

6. Per claim 2

the rejection of claim 1 is incorporated and DaSilva further discloses

- identifying the waypoint in an edited source code during editing of the source code; and compiling the source code up to the identified waypoint before completing the edit of the source code (col.8 lines 55-57).

7. Per claims 3 and 9

the rejection of claims 1 and 8 are incorporated and DaSilva further discloses

- identifying the waypoint includes one of identifying the waypoint from a static definition and identifying the waypoint from a dynamic definition (col.16 lines 20-22 ; set breakpoint at unconditional and conditional break).

8. Per claims 4 and 10

the rejection of claims 1 and 8 are incorporated and DaSilva further discloses

- identifying a second waypoint in the source code during editing of the source code; and compiling the source code from the first waypoint to the second waypoint before completing editing of the source code (col.9 lines 4-6).

9. Per claim 5

the rejection of claim 1 is incorporated and DaSilva further discloses

- completing editing of the source code; and compiling the source code from the second waypoint to the end of the source code (col.9 lines 7-9).

10. Per claim 6

the rejection of claim 1 is incorporated and DaSilva further discloses

- saving the edited source code (col.6 lines 26-28).

11. Per claims 7 and 11

the rejection of claims 1 and 8 are incorporated and DaSilva further discloses

- compiling the source code from the waypoint to the end of the source code upon completing editing of the source code (col.9 lines 7-9).

12. Per claim 8

DaSilva discloses

- A method for use in developing a program, comprising: identifying a waypoint in an edited source code program during editing of the source code program; and compiling the source code program up to the identified waypoint before completing editing of the source code program (col.8 lines 55-57).

13. Per claim 12

DaSilva discloses

- A method for modifying a compiler to engage in rapid compilation, comprising: identifying a file reader portion of the compiler (col.11 lines 51-52); and modifying the

identified file reader to read a portion of a source code program defined by a waypoint from a standard input open (col.18 lines 1-4).

14. Per claim 15

the rejection of claim 12 is incorporated and DaSilva further discloses

- the waypoint is identified by one of identifying the waypoint from a static definition and identifying the waypoint from a dynamic definition (col.16 lines 20-22 ; set breakpoint at unconditional and conditional break).

15. Per claim 16

the rejection of claim 12 is incorporated and DaSilva further discloses

- the waypoint defines a lower bound of the portion of the source code program (col.16 lines 23-33; enter the location where users want to set the breakpoint by setting address or line number).

16. Per claim 17

the rejection of claim 12 is incorporated and DaSilva further discloses

- the waypoint defines an upper bound of the portion of the source code program (col.16 lines 23-33; enter the location where users want to set the breakpoint by setting address or line number).

17. Per claim 18

DaSilva discloses

- A method for suspending compiler execution prior to reaching the end of a source code program, comprising: identifying a waypoint in the source code program (col.16 lines 8-10); compiling a portion of the source code program whose lower bound is defined by the identified waypoint; and suspending compilation of the source code program once the portion whose lower bound is identified by the waypoint is compiled (col.16 lines 23-33; enter the location where users want to set the breakpoint by setting address or line number).

18. Per claim 19

the rejection of claim 18 is incorporated and DaSilva further discloses

- the waypoint is identified by one of identifying the waypoint from a static definition and identifying the waypoint from a dynamic definition (col.16 lines 20-22 ; set breakpoint at unconditional and conditional break).

19. Per claim 20

the rejection of claim 18 is incorporated and DaSilva further discloses

- suspending compilation of the source code program once the portion whose lower bound is identified by the waypoint is compiled includes at least one of removing a corresponding task from a work queue in an IDE (col.16 lines 13-15), storing the compiled code in a shadow location, and suppressing errors or warning (col.6 lines 10-14).

20. Per claim 21

the rejection of claim 18 is incorporated and DaSilva further discloses

- the upper bound of the portion is defined by the start of the source code program or another waypoint (col.16 lines 23-33; enter the location where users want to set the breakpoint by setting address or line number).

21. Per claim 22

DaSilva further discloses

- A method for resuming compiler execution of a suspended compilation, comprising: triggering the compilation of a portion of a source code program whose upper bound is defined by an identified waypoint (col.27 lines 35-37; graph update can be triggered at any point in the program) and compiling the portion of the source code program whose upper bound is defined by the identified waypoint (col.16 lines 23-33; enter the location where users want to set the breakpoint by setting address or line number).

22. Per claim 23

the rejection of claim 22 is incorporated and DaSilva further discloses

- triggering the compilation of the portion of the source code includes identifying the waypoint (col.22 lines 63-67).

23. Per claim 24

DaSilva discloses

- A method for identifying a command and associating it with a file that is being edited (col.2 lines 43-47), comprising: modifying a file reader of a compiler to read from a standard input; and triggering the compilation of a portion of a source code program whose upper bound is defined by an identified waypoint (col.16 lines 23-33; enter the location where users want to set the breakpoint by setting address or line number); invoking the compiler to read the file from the modified file reader through the standard input (col.11 lines 51-52).

24. Per claim 27

the rejection of claim 24 is incorporated and DaSilva further discloses

- triggering the compilation of the portion of the source code includes identifying the waypoint (col.16 lines 8-10).

25. Per claim 28

DaSilva discloses

- A method for building a source code program capable of suspending and resuming compilation, comprising: identifying a waypoint in a source code program being edited (col.2 lines 43-47).
- triggering a compilation of a portion of the source code program defined by the waypoint (col.16 lines 8-10).
- compiling the portion of the source code program defined by the waypoint (col.8 lines 55-57).

- suspending the compilation of the portion defined by the waypoint once the compilation reaches the waypoint; triggering the compilation of the remainder of the source code program; and resuming the compilation of the source code program to compile the remainder (col.16 lines 23-33).

26. Per claim 29

the rejection of claim 28 is incorporated and DaSilva further discloses

- the waypoint is identified by one of identifying the waypoint from a static definition and identifying the waypoint from a dynamic definition (col.16 lines 20-22 ; set breakpoint at unconditional and conditional break).

27. Per claim 30

the rejection of claim 28 is incorporated and DaSilva further discloses

- triggering the compilation of the portion of the source code includes identifying the waypoint (col.22 lines 63-67).

28. Per claim 31

the rejection of claim 28 is incorporated and DaSilva further discloses

- suspending compilation of the source code program once the portion whose lower bound is identified by the waypoint is compiled includes at least one of removing a corresponding task from a work queue in an IDE (col.16 lines 13-15), storing the

compiled code in a shadow location, and suppressing errors or warning (col.6 lines 10-14).

29. Per claim 32

the rejection of claim 28 is incorporated and DaSilva further discloses

- the upper bound of the portion is defined by the start of the source code program or another waypoint (col.16 lines 23-33; enter the location where users want to set the breakpoint by setting address or line number).

30. Per claim 33

the rejection of claim 28 is incorporated and DaSilva further discloses

- triggering the compilation of the remainder of the source code program includes identifying a second waypoint, saving the source code program, or ending an editing session (col.22 lines 63-67).

31. Per claim 38

DaSilva discloses

- A method for managing the output of a compile, comprising: compiling at least a portion of a source code program defined by a waypoint during the editing of the source code program in a first phase (col.2 lines 50-55; setting a breakpoint to stop the execution of a program).

- compiling the remainder of the source code program in a subsequent phase (col.9 lines 7-9)
- notifying a user of any errors that may have occurred during the compilation (col.6 lines 10-14).

Per claim 39

the rejection of claim 38 is incorporated and DaSilva further discloses

- the portion comprises a portion of the source code program defined by the start of the source code program and the waypoint (col.16 lines 23-33; enter the location where users want to set the breakpoint by setting address or line number).

32. Per claim 40

the rejection of claim 38 is incorporated and DaSilva further discloses

- the portion comprises a portion of the source code program defined by the waypoint and the end of the source code program (col.16 lines 23-33; enter the location where users want to set the breakpoint by setting address or line number).

33. Per claim 41

the rejection of claim 38 is incorporated and DaSilva further discloses

- the waypoint is identified by one of identifying the waypoint from a static definition and identifying the waypoint from a dynamic definition (col.16 lines 20-22 ; set breakpoint at unconditional and conditional break).

34. Per claim 42

the rejection of claim 38 is incorporated and DaSilva further discloses

- comprising scrapping the compiled first and second portions (col.9 lines 4-6).

35. Per claim 43

the rejection of claim 42 is incorporated and DaSilva further discloses

- scrapping the compiled first and second portions includes one of scrapping the compiled first and second portions responsive to the notification and scrapping the compiled first and second portions responsive to a user input (col.16 lines 13-15).

36. Per claim 44

DaSilva discloses

- A method for use in developing a program, comprising: identifying at least two or more instructions in a file to compile; and compiling the identified instructions while the file is being edited (col.8 lines 61-67; run program by step through a single instruction or a function).

37. Per claim 45

the rejection of claim 44 is incorporated and DaSilva further discloses

- The instructions are identified at a predetermined line number in the source code program, identifying the instructions at the point of insertion for a text editor, identifying

the instructions after a predetermined number of branches as conditionals, identifying the instructions at a predetermined text offset (col.18 lines 22-27)

38. Per claim 46

the rejection of claim 44 is incorporated and DaSilva further discloses

- identifying at least two more instructions in the file during editing; and compiling the second two or more instruction while the file is being edited (col.8 lines 61-67; run program by step through a single instruction or a function).

39. Per claim 47

the rejection of claim 44 is incorporated and DaSilva further discloses

- completing editing of the file; and compiling the remainder of the edited file (col.9 lines 7-9).

40. Per claim 48

the rejection of claim 44 is incorporated and DaSilva further discloses

- comprising saving the edited file (col.6 lines 26-28).

41. Per claim 49

the rejection of claim 44 is incorporated and DaSilva further discloses

- compiling the remainder of the edited file upon completing editing of the file (col.9 lines 7-9).

42. Per claim 50

- A method for compiling a source code program, comprising: identifying an upper bound for a portion of the source code program to compile; identifying a lower bound for the portion; and compiling the portion defined by the upper and lower bounds during an editing session on the source code program (col.16 lines 23-33; enter the location where users want to set the breakpoint by setting address or line number).

43. Per claim 51

the rejection of claim 50 is incorporated and DaSilva further discloses

- at least one of identifying the upper bound and identifying the lower bound includes one of identifying the bound from a static definition and identifying the bound from a dynamic definition (col.16 lines 20-22 ; set breakpoint at unconditional and conditional break).

44. Per claim 52

the rejection of claim 50 is incorporated and DaSilva further discloses

- identifying a third bound in the edited source code during editing of the source code; and compiling the source code from the lower bound to the third bound before completing editing of the source code (col.12 lines 51-54; program runs from last breakpoint to the next breakpoint).

45. Per claim 53

the rejection of claim 50 is incorporated and DaSilva further discloses

- compiling the source code from the lower bound to the end of the source code upon completing editing of the source code (col.9 lines 7-9).

Claim Rejections - 35 USC § 103

46. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

47. Claims 13,14,25, and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over DaSilva, in view of Meth (U.S. Pub No. 20020087916 A1).

48. Per claims 13 and 25

the rejection of claim 12 is incorporated

But DaSilva does not include

- modifying the identified file reader to read from the standard input includes modifying the identified file reader to read from an open system call.

However Meth discloses

- modifying the identified file reader to read from the standard input includes modifying the identified file reader to read from an open system call ([0039] *"whenever the program opens a file with the open( ) system call, the Condor user-level checkpoint mechanism intercepts the open() system call and records for itself the name of the file being opened"*).
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify teaching of DaSilva with the teachings of Meth to include modifying the identified file reader to read from the standard input includes modifying the identified file reader to read from an open system call in order to use to open and close files whenever the program opens a file with the open system call, the user-level checkpoint mechanism intercepts the open system call and records for itself the name of the file being open (see [0039]).

49. Per claims 14 and 26

the rejection of claim 13 is incorporated and Meth further discloses

- modifying the identified file reader to read from the open system call includes modifying the identified file reader to read from a UNIX gcc command [0039] *"... via standard UNIX system calls..."* which is implicitly included gcc command).

50. Claims 34-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over DaSilva, in view of Sollich (U.S. Pub No. 20020016953 A1).

51. Per claim 34

DaSilva discloses

- A method for using a UNIX standard input read mechanism for speculative compilation of a source code program, comprising: identifying a waypoint in an edited source code program during editing of the source code program (col.8 lines 55-57);

But DaSilva does not disclose

- invoking a compile of at least a portion of a source code program defined by a waypoint during the editing of the source code program with a UNIX input read mechanism.

However Sollich discloses

- invoking a compile of at least a portion of a source code program defined by a waypoint during the editing of the source code program with a UNIX input read mechanism (Sollich [0061] *"...the Integrated Development Environment or IDE invokes the compiler for determining an appropriate context for the source code, based on where the screen cursor is currently positioned within the code."*
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify teaching of DaSilva with the teachings of Sollich to include invoking a compile of at least a portion of a source code program defined by a waypoint during the editing of the source code program with a UNIX input read mechanism in order to get a result to the IDE which describes the current context within the source code from compiler (see [0061]).

52. Per claim 35

the rejection of claim 34 is incorporated and DaSilva further discloses

- the portion comprises a portion of the source code program defined by the start of the source code program and the waypoint (col.16 lines 23-33; enter the location where users want to set the breakpoint by setting address or line number).

53. Per claim 36

the rejection of claim 34 is incorporated and DaSilva further discloses

- the portion comprises a portion of the source code program defined by the waypoint and the end of the source code program (col.16 lines 23-33; enter the location where users want to set the breakpoint by setting address or line number).

54. Per claim 37

the rejection of claim 34 is incorporated and DaSilva further discloses

- the waypoint is identified by one of identifying the waypoint from a static definition and identifying the waypoint from a dynamic definition (col.16 lines 20-22 ; set breakpoint at unconditional and conditional break).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Junchun Wu whose telephone number is 571-270-1250. The examiner can normally be reached on 8:00-17:00 M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JW

WEI ZHEN
SUPERVISORY PATENT EXAMINER